A Deterministic Polynomial–Time Algorithm for Constructing a Multicast Coding Scheme for Linear Deterministic Relay Networks

S. M. Sadegh Tabatabaei Yazdi and Serap A. Savari*

Abstract

We propose a new way to construct a multicast coding scheme for linear deterministic relay networks. Our construction can be regarded as a generalization of the well-known multicast network coding scheme of Jaggi et al. to linear deterministic relay networks and is based on the notion of flow for a unicast session that was introduced by the authors in earlier work. We present randomized and deterministic polynomial–time versions of our algorithm and show that for a network with g destinations, our deterministic algorithm can achieve the capacity in $\lceil \log(g+1) \rceil$ uses of the network.

1 Introduction

Computing the capacity and constructing optimal coding schemes for wireless Gaussian networks are central open questions and of great importance in network information theory. In a wireless network the transmitted signal from a node is broadcasted to all its neighbors and the signal received at a node is the superposition of the signals transmitted by its neighbors and Gaussian noise. Broadcasting, interference, and noise are the three main characteristics of a wireless network that differentiate it from a wired network and make its analysis much more challenging. Recently Avestimehr, Diggavi, and Tse [2] proposed an approximation model known as the *linear deterministic relay network* (LDRN) for wireless Gaussian networks that simplifies the three features of wireless Gaussian networks by instead considering deterministic and linear operations in vector spaces over finite fields. Avestimehr, Diggavi, and Tse have further shown that for some Gaussian wireless networks, the capacity of the wireless network is within an additive constant gap from the capacity of the corresponding approximation network and the optimal coding scheme for the approximation network can be translated to near optimal coding schemes for the Gaussian wireless network [3].

An LDRN is a wireless networking model which can be visualized as a layered directed network $\mathcal{N} = (V, E)$ with set of "nodes" $V = \bigcup_{i=1}^{M} V_i$, where V_i denotes the set of nodes in layer i, and set of "edges" E. Let $V_i = \{v_i(1), \cdots, v_i(m_i)\}$, where m_i denotes the number of nodes in layer i. The first layer consists of a single node $s = v_1(1)$ called the source node. There are g destination nodes denoted by $t_l \triangleq v_{K_l}(d_l), l \in \{1, \cdots, g\}$, distributed in layers $K_1, K_2, \cdots, K_g$. There is an "edge" from every node in V_i to every node in V_{i+1} which corresponds to the *transfer matrix* between the two nodes. Figure 1 is an example of an LDRN with four layers and two destination nodes.

*The authors are with the Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843 USA. Their e-mail addresses are sadegh@neo.tamu.edu and savari@ece.tamu.edu.

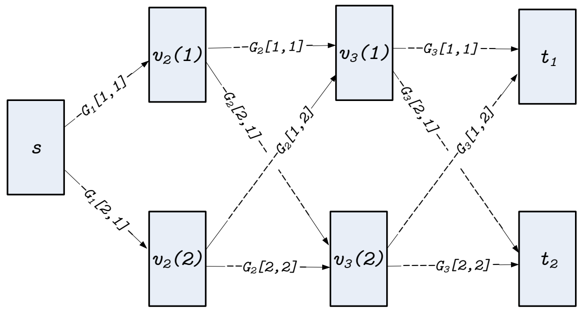

Figure 1: An LDRN with four layers. Here $t_1 = v_4(1)$ and $t_2 = v_4(2)$.

During one use of the communication channel between layers i and $i+1$, $v_i(j)$ transmits a predetermined length vector $\mathbf{x}_i[j]$ to the nodes in layer $i+1$ and $v_{i+1}(k)$ receives a predetermined length vector $\mathbf{y}_{i+1}[k]$ given by

$$\mathbf{y}_{i+1}[k] = \sum_{j=1}^{m_i} G_i[k,j]\mathbf{x}_i[j],$$

where $G_i[k,j]$ is a predetermined transfer matrix of the edge $(v_i(j), v_{i+1}(k)) \in E$. Note that we can set $G_i[k,j]$ to be the all-zero matrix if there is no connection from $v_i(j)$ to $v_{i+1}(k)$. All vectors and matrices are over a fixed finite field $\mathbb{F}$. One can define

$$\mathbf{x}_i = \begin{bmatrix} \mathbf{x}_i[1] \\ \vdots \\ \mathbf{x}_i[m_i] \end{bmatrix}, \mathbf{y}_{i+1} = \begin{bmatrix} \mathbf{y}_{i+1}[1] \\ \vdots \\ \mathbf{y}_{i+1}[m_{i+1}] \end{bmatrix}$$

and the block matrix $G_i = [G_i[k,j]], 1 \le k \le m_{i+1}, 1 \le j \le m_i$. Then the received vectors at layer $i+1$ are related to the transmitted vectors at layer i by the following relationship

$$\mathbf{y}_{i+1} = G_i\mathbf{x}_i.$$

The capacity of an LDRN for a single multicast session from source s to the destinations $t_1, \cdots, t_g$ is derived in [2]. Define a cut between the source node s and a destination node t_j as a partition of nodes V into two sets A and B, with $s \in A$ and $t_j \in B$. The capacity of the cut is defined as the rank of the transfer matrix from the transmitted vectors of the nodes in A to the received vectors of the nodes in B. [2] shows that the minimum capacity of the cuts between s and t_j is the capacity of a unicast session between s and t_j. Furthermore the multicast capacity of the network between source s and destinations $t_1, \cdots, t_g$ is the minimum of the min–cut capacities between the source and each destination. The capacity–achieving scheme in [2] is a random linear coding scheme that is asymptotically optimal when the network is used for multiple rounds.

A few groups of researchers (see, e.g., [1, 6, 11, 12]) have proposed deterministic coding schemes for the transmission of a single unicast session over an LDRN which can be constructed in polynomial time. Furthermore, they achieve capacity using only one round of the network. These schemes are similar to routing schemes in wired networks and have low encoding and decoding complexities at the relay nodes.

In this paper we build upon our work in [11, 12] to design a simple and low complexity transmission scheme for a multicast session over an LDRN. Our scheme will be constructed by progressively combining the coding schemes for unicast sessions from the source to each destination. In many

ways our scheme is similar to and is a generalization of the scheme in [7] for a multicast session in wired networks. We will offer both randomized and deterministic versions of our algorithm and show that $\lceil \log(g+1) \rceil$ uses of the network suffice to achieve capacity, which resembles the result for wired networks [7].

For the case of a single multicast session, there have been multiple recent attempts to devise deterministic and efficient algorithms for constructing capacity–achieving coding schemes. In [4], Ebrahimi and Fragouli developed an algebraic framework for vector network coding and used this framework to devise a multicast transmission scheme over an LDRN. Our scheme has a lower complexity of construction and needs fewer uses of the network to achieve capacity. Erez et al. [5] offer a different construction by progressing through the network according to a topological order and maintaining the linear independence of certain subsets of coding vectors along the processing. However, the proposed algorithm does not appear to have a polynomial running time. Kim and Médard [9] generalized the algebraic framework of Koetter and Médard [10] for classical network coding to LDRNs and devised an algebraic algorithm for constructing multicast codes. Again, the proposed algorithm does not appear to have a polynomial running time. More recently, [8] proposed an algorithm using rotational codes to asymptotically achieve the multicast capacity of LDRN networks for a multicast session. Rotational codes have some built–in advantages as they are easy to implement at the relay nodes. However, the existence of deterministic polynomial–time algorithms for the construction of efficient rotational codes for multicast transmission over an LDRN remains unknown.

We will next review our earlier results on a single unicast session [11, 12] in Section 2 and then discuss our coding construction for a multicast session in Section 3.

2 A single unicast session

In this section we briefly explain the coding scheme for a single unicast session from [11, 12]. This will be the building block of our multicast coding scheme.

Recall that for each $i \in \{1, \cdots, M-1\}$ the transmitted vector of layer i and the received vector of layer $i+1$ are related through matrix G_i by $\mathbf{y}_{i+1} = G_i \mathbf{x}_i$.

For each layer $i \in \{1, \cdots, M\}$ label the indices of the vector $\mathbf{y}_i$ with the elements of a set P_i and label the indices of the vector $\mathbf{x}_i$ with the elements of a set Q_i. We choose all sets P_i and Q_i to be disjoint for different values of i. For any $A \subseteq P_i$, let $\mathbf{y}_i(A)$ denote the subvector of $\mathbf{y}_i$ corresponding to indices with labels from set A. Similarly, for any $B \subseteq Q_i$, let $\mathbf{x}_i(B)$ denote the subvector of $\mathbf{x}_i$ associated with indices with labels from set B. Next partition each set P_i into subset $P_i = \cup_{j=1}^{m_i} P_i[j]$ and Q_i into subsets $Q_i = \cup_{j=1}^{m_i} Q_i[j]$ such that $P_i[j]$ is the subset of indices of $\mathbf{y}_i$ that belong to the subvector $\mathbf{y}_i[j]$ and $Q_i[j]$ is the subset of indices of $\mathbf{x}_i$ that belong to the subvector $\mathbf{x}_i[j]$. Therefore we have $\mathbf{y}_i[j] = \mathbf{y}_i(P_i[j])$ and $\mathbf{x}_i[j] = \mathbf{x}_i(Q_i[j])$ for any $j \in \{1, \cdots, m_i\}$. For any $i \in \{1, \cdots, M-1\}$ we will use the sets P_{i+1} and Q_i to label the rows and the columns of the matrix G_i such that for each $p \in P_{i+1}$ the row of G_i corresponding to the element $\mathbf{y}_{i+1}(p)$ is labeled with p and for each $q \in Q_i$ the column of G_i corresponding to the element $\mathbf{x}_i(q)$ is labeled with q. For $p \in P_{i+1}$ and $q \in Q_i$ let $G_i(p, q)$ denote the element in row p and column q of matrix G_i. For $A \subseteq P_{i+1}$ and $B \subseteq Q_i$ let $G_i(A, B)$ denote the submatrix of G_i consisting of the rows in A and the columns in B. Our labeling implies that $G_i(P_{i+1}[k], Q_i[j]) = G_i[k, j]$ for any $j \in \{1, \cdots, m_i\}$ and $k \in \{1, \cdots, m_{i+1}\}$.

If node s holds a column vector message $\mathbf{w} \in \mathbb{F}^{R \times 1}$ and we are looking at a linear coding scheme, then at each layer $i \in \{1, \cdots, M\}$, each element of vectors $\mathbf{x}_i$ and $\mathbf{y}_i$ will be a linear transformation of the vector $\mathbf{w}$. We represent the "global coding vector" (see [7]) for the element $\mathbf{x}_i(q), q \in Q_i$, with row vector $\boldsymbol{x}_i(q) \in \mathbb{F}^{1 \times R}$ such that $\mathbf{x}_i(q) = \boldsymbol{x}_i(q)\mathbf{w}$ and the global coding vector for the element

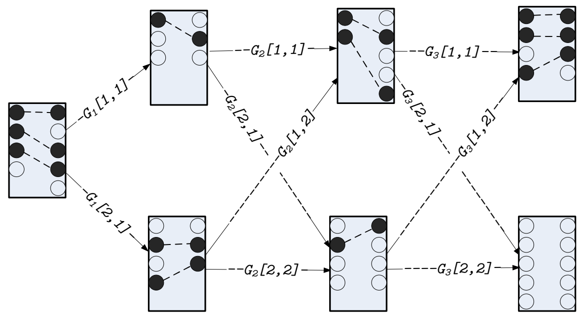

Figure 2: An example of a rate–3 flow from the source node s to the destination node t_1. Here the matched elements of flow are connected together through dashed lines.

$\mathbf{y}_i(p), p \in P_i$, with row vector $\boldsymbol{y}_i(p) \in \mathbb{F}^{1\times R}$ such that $\mathbf{y}_i(p) = \boldsymbol{y}_i(p)\mathbf{w}$. For subsets $B \subseteq Q_i$ and $A \subseteq P_i$ we use the notation $\boldsymbol{x}_i(B)$ and $\boldsymbol{y}_i(A)$ to respectively denote the matrices that are formed by the vectors $\boldsymbol{x}_i(q)$ and $\boldsymbol{y}_i(p)$ for $q \in B$ and $p \in A$. Therefore we have $\mathbf{x}_i(B) = \boldsymbol{x}_i(B)\mathbf{w}$ and $\mathbf{y}_i(A) = \boldsymbol{y}_i(A)\mathbf{w}$.

Suppose that the network supports a rate–R unicast connection between source node s and a destination node $t = v_K(d)$ for $K \leq M$ and $d \in \{1, \cdots, m_K\}$. The main result of [11, 12] can be summarized in the following theorem:

Theorem 1. *For each $1 \leq i \leq K$ and for each $1 \leq j, k \leq m_i$ there exist subsets $\hat{Q}_i[j] \subseteq Q_i[j]$ and $\hat{P}_i[k] \subseteq P_i[k]$ such that the following hold*

1. $|\hat{P}_i[j]| = |\hat{Q}_i[j]|$ *for* $i \in \{1, \cdots, K\}, j \in \{1, \cdots, m_i\}$,
2. $\sum_{j=1}^{m_i} |\hat{P}_i[j]| = \sum_{j=1}^{m_i} |\hat{Q}_i[j]| = R$, *for* $i \in \{1, \cdots, K-1\}$,
3. $|\hat{P}_K[d]| = R$ *and* $|\hat{P}_K[k]| = 0$ *for* $k \neq d$,
4. $G_i(\bigcup_{k=1}^{m_{i+1}} \hat{P}_{i+1}[k], \bigcup_{j=1}^{m_i} \hat{Q}_i[j])$ *is a nonsingular matrix for* $i \in \{1, \cdots, K-1\}$.

Furthermore such subsets can be found by an algorithm that runs in a time that is polynomial in the size of the network $\mathcal{N}$.

We call the subsets $\hat{Q}_i[j] \subseteq Q_i[j]$ and $\hat{P}_i[k] \subseteq P_i[k]$ for $i \in \{1, \cdots, M\}$ and $j, k \in \{1, \cdots, m_i\}$ a *flow* of rate R in the LDRN from the source node s to the destination node t.

The four properties of a flow in Theorem 1 depend on $G_1, ..., G_{M-1}$ and do not depend on the specific choice of the set $\hat{P}_1[1]$ among all subsets of $P_1[1]$ with size R. Therefore, if there exists a rate–R flow, we can set $\hat{P}_1[1]$ to be any subset of $P_1[1]$ of size R.

Notice that the existence of a flow of rate R implies the following simple and low complexity coding scheme of rate R from the source s to the destination t: To send message $\mathbf{w} \in \mathbb{F}^{R\times 1}$, source node $s = v_1(1)$ sets $\mathbf{y}_1(\hat{P}_1[1]) = \mathbf{w}$ and $\mathbf{y}_1(P_1[1] \setminus \hat{P}_1[1]) = \mathbf{0}$. Next, any node $v_i(j), i \in \{1, \cdots, M\}, j \in \{1, \cdots, m_i\}$, in the network forms the vector $\mathbf{x}_i[j]$ by setting

$$\mathbf{x}_i(\hat{Q}_i[j]) = \mathbf{y}_i(\hat{P}_i[j]).$$

We say that element $p \in \hat{P}_i[j]$ is "matched" with element $q \in \hat{Q}_i[j]$ when $\mathbf{x}_i(q)$ is set to $\mathbf{y}_i(p)$ through the preceding equation (see Figure 2 for an example of a flow). We further let $\mathbf{x}_i(Q_i[j] \setminus \hat{Q}_i[j]) = \mathbf{0}$.

From the properties of flow it follows that at the destination $t = v_K(d)$

$$\mathbf{x}_K(\hat{Q}_K[d]) = G_{K-1}(\hat{P}_K[d], \bigcup_{j=1}^{m_{K-1}} \hat{Q}_{K-1}[j]) \cdots G_2(\bigcup_{k=1}^{m_3} \hat{P}_3[k], \bigcup_{j=1}^{m_2} \hat{Q}_2[j]) G_1(\bigcup_{k=1}^{m_2} \hat{P}_2[k], \hat{Q}_1[1])\mathbf{w}.$$

Since each matrix $G_i(\bigcup_{k=1}^{m_{i+1}} \hat{P}_{i+1}[k], \bigcup_{j=1}^{m_i} \hat{Q}_i[j])$ is nonsingular, node t can recover vector $\mathbf{w}$ from the received vector $\mathbf{x}_K(\hat{Q}_K[d])$ through a linear transformation.

3 A coding scheme for a multicast session

Assume that there are g destination nodes $t_1, \cdots, t_g$ in the network and the min–cut capacity from the source node s to each destination is at least R. We are interested in a multicast coding scheme in which all destinations can simultaneously receive the message $\mathbf{w} \in \mathbb{F}^{R\times 1}$ of the source. Our scheme will be designed by combining the flows of rate R from the source to each destination.

Suppose that $t_l = v_{K_l}(d_l)$ for $l \in \{1, \cdots, g\}$. From Theorem 1 for each $t_l, l \in \{1, \cdots, g\}$, there exists a flow with subsets $P_i^l[k] \subseteq P_i[k]$ and $Q_i^l[j] \subseteq Q_i[j]$ for $1 \leq i \leq K_l$ and $1 \leq j, k \leq m_i$ such that:

1. $|P_i^l[j]| = |Q_i^l[j]|$ for $i \in \{1, \cdots, K_l\}, j \in \{1, \cdots, m_i\}$,
2. $\sum_{j=1}^{m_i} |P_i^l[j]| = \sum_{j=1}^{m_i} |Q_i^l[j]| = R$, for $i \in \{1, \cdots, K_l - 1\}$,
3. $|P_{K_l}^l[d_l]| = R$ and $|P_{K_l}^l[k]| = 0$ for $k \neq d_l$,
4. $G_i(\bigcup_{k=1}^{m_{i+1}} P_{i+1}^l[k], \bigcup_{j=1}^{m_i} Q_i^l[j])$ is a nonsingular matrix for $i \in \{1, \cdots, K_l - 1\}$.

Since $P_1^l[1], l \in \{1, \cdots, g\}$, can be any subset of $P_1[1]$ of size R, we set all subsets $P_1^l[1], l \in \{1, \cdots, g\}$, to be the same subset of $P_1[1]$.

Our design criterion for a multicast coding scheme is that for each destination $t_l, l \in \{1, \cdots, g\}$, at each layer $i \in \{1, \cdots, K_l\}$, the global coding vectors correponding to the elements of the vectors $\mathbf{y}_i(P_i^l[j])$ for $j \in \{1, \cdots, m_i\}$ must be linearly independent vectors and hence the length–R vector

$$\mathbf{y}_i \left(\bigcup_{j=1}^{m_i} P_i^l[j] \right)$$

can uniquely determine the message vector $\mathbf{w}$. In other words we require for each destination t_l and each layer $i \in \{1, \cdots, K_l\}$:

- Condition (*): the matrix $\boldsymbol{y}_i \left(\bigcup_{j=1}^{m_i} P_i^l[j] \right)$ must be nonsingular.

The destination node $t_l = v_{K_l}(d_l)$ will receive the length–R vector $\mathbf{y}_{K_l}(P_{K_l}^l[d_l]) = \boldsymbol{y}_{K_l}(P_{K_l}^l[d_l])\mathbf{w}$. Since $\boldsymbol{y}_{K_l}(P_{K_l}^l[d_l])$ is a nonsingular matrix, t_l will be able to decode message $\mathbf{w}$.

Notice that at each node $v_i(j)$ for $i \in \{2, \cdots, M\}$ we only have control over the design of the coding vectors $\boldsymbol{x}_i(q)$ for $q \in Q_i[j]$ which can be a linear function of the coding vectors $\{\boldsymbol{y}_i(p) : p \in P_i[j]\}$. The coding vectors $\boldsymbol{y}_i(p)$ for $p \in P_i[j]$ are determined from the coding vectors of the previous layer and matrix G_{i-1}. In our design we will assign coding vectors layer by layer, starting from the first layer. At each layer i we fix an arbitrary order on the elements of the set Q_i and assign the coding vectors $\boldsymbol{x}_i(q)$ in this order.

Initialization: We start from the first layer. Since $P_1^l[1]$ is the same subset for every $l \in \{1, \cdots, g\}$ we set $\boldsymbol{y}_1(P_1^l[1]) = I_{R\times R}$, i.e., the $R \times R$ identity matrix, and set $\boldsymbol{y}_1(P_1[1] \setminus P_1^l[1]) = \mathbf{0}$ for every $l \in \{1, \cdots, g\}$. In other words we set $\mathbf{y}_1(P_1^l[1]) = \mathbf{w}$ and $\mathbf{y}_1(P_1[1] \setminus P_1^l[1]) = \mathbf{0}$ for every $l \in \{1, \cdots, g\}$. Therefore condition (*) will be satisfied for all destinations in the first layer.

Inductive Step: We continue our coding construction inductively. Suppose that the condition (*) holds for layer i and for all destinations $t_l = v_{K_l}(d_l)$ with $K_l \geq i$. Next we will design the coding vectors $\boldsymbol{x}_i(q)$ for $q \in Q_i$ one by one and in the order of the elements of Q_i so that at the end the condition (*) holds for layer $i+1$ and all destinations $t_l = v_{K_l}(d_l)$ with $K_l \geq i+1$.

At this step of the algorithm for each destination t_l with $K_l \geq i+1$ we maintain two matrices. One is the matrix A_l which is initially

$$A_l = \boldsymbol{y}_i \left(\bigcup_{j=1}^{m_i} P_i^l[j] \right),$$

and is updated throughout the algorithm. The other matrix is

$$F_l = G_i(\bigcup_{k=1}^{m_{i+1}} P_{i+1}^l[k], Q_l'),$$

where initially $Q_l' = \bigcup_{j=1}^{m_i} Q_i^l[j]$ and it is updated throughout the algorithm. Throughout the algorithm we maintain the invariance that the product $F_l A_l$ is a nonsingular matrix for every destination t_l with $K_l \geq i+1$. We will also verify that after all of the elements of Q_i are processed, for every destination t_l with $K_l \geq i+1$ we will have $F_l A_l = \boldsymbol{y}_{i+1}\left(\bigcup_{j=1}^{m_{i+1}} P_{i+1}^l[j]\right)$, which is sufficient for condition (*) to hold at layer $i+1$.

A_l is initially invertible since condition (*) holds for layer i. Matrix F_l is also initially nonsingular by the definition of a flow to destination t_l given in Theorem 1. Therefore the product $F_l A_l$ is initially nonsingular. Next we will explain the design of the coding vector $\boldsymbol{x}_i(q)$ for $q \in Q_i$ and describe the updating process of F_l and A_l for every destination t_l with $K_l \geq i+1$. We consider two cases:

1. If q is part of the flow for destination t_l, i.e., $q \in Q_i^l[j]$ for some $j \in \{1, \cdots, m_i\}$, then update matrix A_l by replacing row $\boldsymbol{y}_i(p_l)$ with $\boldsymbol{x}_i(q)$, which we will later explain how to design. Here $p_l \in P_i^l[j]$ is the unique element that is matched with $q \in Q_i^l[j]$ in the flow for destination t_l. There is no change needed for matrix F_l.

2. If q is not part of the flow for destination t_l, then update A_l adding a new row $\boldsymbol{x}_i(q)$ to it and insert a column into F_l so that the set of column indices grows from Q_l' to $Q_l' \cup \{q\}$. In this step we place $\boldsymbol{x}_i(q)$ in the row of A_l counting from the top which is the same as the position of the new column $G_i(\bigcup_{k=1}^{m_{i+1}} P_{i+1}^l[k], \{q\})$ in the updated F_l counting from the left.

When we have gone through all of the elements of Q_i, matrix F_l would be $G_i(\bigcup_{k=1}^{m_{i+1}} P_{i+1}^l[k], Q_i)$ and matrix A_l would be the matrix $\boldsymbol{x}_i(Q_i)$. Therefore we have

$$F_l A_l = G_i(\bigcup_{k=1}^{m_{i+1}} P_{i+1}^l[k], Q_i)\boldsymbol{x}_i(Q_i) = \boldsymbol{y}_{i+1}\left(\bigcup_{j=1}^{m_{i+1}} P_{i+1}^l[j]\right)$$

where the second equation holds since G_i is the transfer matrix from $\boldsymbol{x}_i(Q_i) = \boldsymbol{x}_i$ to $\boldsymbol{y}_{i+1}$. This equation guarantees that $\boldsymbol{y}_{i+1}\left(\bigcup_{j=1}^{m_{i+1}} P_{i+1}^l[j]\right)$ is nonsingular, as desired.

Next we analyze each case and find the condition that $\boldsymbol{x}_i(q)$ needs to satisfy in order for $F_l A_l$ to remain nonsingular:

Analysis of Case 1

Without loss of generality suppose that $\boldsymbol{x}_i(q)$ is the first row of A_l and that matrix A_l after the update is of the form

$$A_l = \begin{bmatrix} \boldsymbol{x}_i(q) \\ A'_l \end{bmatrix}.$$

Therefore A_l before the update is of the form $\begin{bmatrix} \boldsymbol{y}_i(p_l) \\ A'_l \end{bmatrix}$. We require that the matrix F_lA_l be nonsingular. We write

$$F_l = \begin{bmatrix} \boldsymbol{\alpha} & F'_l \end{bmatrix}$$

where $\boldsymbol{\alpha} \in \mathbb{F}^{R\times 1}$ is the first column of F_l. Using standard matrix calculus we can write

$$\begin{aligned} F_lA_l &= \begin{bmatrix} \boldsymbol{\alpha} & F'_l \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_i(q) \\ A'_l \end{bmatrix} \\ &= \boldsymbol{\alpha}\boldsymbol{x}_i(q) + F'_lA'_l. \end{aligned}$$

Let us define $H = \begin{bmatrix} \boldsymbol{\alpha} & F'_l \end{bmatrix} \begin{bmatrix} \boldsymbol{y}_i(p_l) \\ A'_l \end{bmatrix}$, which is the matrix F_lA_l resulting from the previous step and is nonsingular by the inductive assumption. We can write

$$F'_lA'_l = H - \boldsymbol{\alpha}\boldsymbol{y}_i(p_l)$$

and therefore

$$F_lA_l = H + \boldsymbol{\alpha}(\boldsymbol{x}_i(q) - \boldsymbol{y}_i(p_l)).$$

For the moment suppose that F_lA_l is singular. This means that there exist a non–zero column vector $\boldsymbol{\beta} \in \mathbb{F}^{R\times 1}$ with $F_lA_l\boldsymbol{\beta} = \mathbf{0}$. This implies that

$$H\boldsymbol{\beta} + \boldsymbol{\alpha}(\boldsymbol{x}_i(q) - \boldsymbol{y}_i(p_l))\boldsymbol{\beta} = \mathbf{0}. \tag{1}$$

Since H is a nonsingular matrix, there is a vector $\boldsymbol{\gamma}_l$ such that $\boldsymbol{\alpha} = H\boldsymbol{\gamma}_l$. Then (1) can be rewritten as

$$H\boldsymbol{\beta} + H\boldsymbol{\gamma}_l(\boldsymbol{x}_i(q) - \boldsymbol{y}_i(p_l))\boldsymbol{\beta} = H(\boldsymbol{\beta} + \boldsymbol{\gamma}_l(\boldsymbol{x}_i(q) - \boldsymbol{y}_i(p_l))\boldsymbol{\beta}) = \mathbf{0}.$$

Since H is nonsingular, the identity holds if and only if

$$\boldsymbol{\beta} + \boldsymbol{\gamma}_l(\boldsymbol{x}_i(q) - \boldsymbol{y}_i(p_l))\boldsymbol{\beta} = \mathbf{0}.$$

If we premultiply the vectors from both sides of the preceding vector equation by $(\boldsymbol{x}_i(q) - \boldsymbol{y}_i(p_l))$, we find that

$$(\boldsymbol{x}_i(q)-\boldsymbol{y}_i(p_l))\boldsymbol{\beta}+(\boldsymbol{x}_i(q)-\boldsymbol{y}_i(p_l))\boldsymbol{\gamma}_l(\boldsymbol{x}_i(q)-\boldsymbol{y}_i(p_l))\boldsymbol{\beta} = (1+(\boldsymbol{x}_i(q)-\boldsymbol{y}_i(p_l))\boldsymbol{\gamma}_l)(\boldsymbol{x}_i(q)-\boldsymbol{y}_i(p_l))\boldsymbol{\beta} = 0.$$

The expression above is product of two numbers $(1 + (\boldsymbol{x}_i(q) - \boldsymbol{y}_i(p_l))\boldsymbol{\gamma}_l)$ and $(\boldsymbol{x}_i(q) - \boldsymbol{y}_i(p_l))\boldsymbol{\beta}$. We argue that $(\boldsymbol{x}_i(q) - \boldsymbol{y}_i(p_l))\boldsymbol{\beta}$ is not zero. Observe that if this number was zero, then equation (1) and the nonsingularity of H would imply that $H\boldsymbol{\beta}$ and $\boldsymbol{\beta}$ are both zero vectors, contradicting our assumption that $\boldsymbol{\beta}$ is a non–zero vector. Therefore

$$1 + (\boldsymbol{x}_i(q) - \boldsymbol{y}_i(p_l))\boldsymbol{\gamma}_l = 0.$$

This argument implies that for F_lA_l to be nonsingular it is sufficient to have the following inequality:

$$1 + (\boldsymbol{x}_i(q) - \boldsymbol{y}_i(p_l))\boldsymbol{\gamma}_l \neq 0. \tag{2}$$

Analysis of Case 2

The analysis is very similar to Case 1. Without loss of generality assume that the new row is added to the bottom of A_l and the new column is added to the right of F_l. After the update A_l is of the form

$$A_l = \begin{bmatrix} A_l' \\ \boldsymbol{x}_i(q) \end{bmatrix}.$$

Here A_l' represents matrix A_l before the update. Also matrix F_l after the update is of the form

$$F_l = \begin{bmatrix} F_l' & \boldsymbol{\alpha} \end{bmatrix}$$

where $\boldsymbol{\alpha} \in \mathbb{F}^{R\times 1}$ is the new column added to F_l', which is the matrix F_l before the update. Our inductive assumption implies that $H = F_l' A_l'$ is nonsingular. We can write

$$F_l A_l = H + \boldsymbol{\alpha}\boldsymbol{x}_i(q).$$

$F_l A_l$ is singular if there exists a non–zero vector $\boldsymbol{\beta}$ such that

$$F_l A_l \boldsymbol{\beta} = H\boldsymbol{\beta} + \boldsymbol{\alpha}\boldsymbol{x}_i(q)\boldsymbol{\beta} = \boldsymbol{0}.$$

Since H is nonsingular, there exists a vector $\boldsymbol{\gamma}_l$ such that $\boldsymbol{\alpha} = H\boldsymbol{\gamma}_l$. Therefore FA_l is singular if there exists a non–zero vector $\boldsymbol{\beta}$ for which

$$H\boldsymbol{\beta} + H\boldsymbol{\gamma}_l\boldsymbol{x}_i(q)\boldsymbol{\beta} = H(\boldsymbol{\beta} + \boldsymbol{\gamma}_l\boldsymbol{x}_i(q)\boldsymbol{\beta}) = \boldsymbol{0}. \tag{3}$$

Since H is nonsingular, (3) implies

$$\boldsymbol{\beta} + \boldsymbol{\gamma}_l\boldsymbol{x}_i(q)\boldsymbol{\beta} = \boldsymbol{0}.$$

If we premultiply both sides of the preceding equation by $\boldsymbol{x}_i(q)$ we obtain

$$\boldsymbol{x}_i(q)\boldsymbol{\beta} + \boldsymbol{x}_i(q)\boldsymbol{\gamma}_l\boldsymbol{x}_i(q)\boldsymbol{\beta} = \boldsymbol{x}_i(q)\boldsymbol{\beta}(1 + \boldsymbol{x}_i(q)\boldsymbol{\gamma}_l) = 0.$$

The previous equality holds if either $\boldsymbol{x}_i(q)\boldsymbol{\beta} = 0$ or if $(1 + \boldsymbol{x}_i(q)\boldsymbol{\gamma}_l) = 0$. If $\boldsymbol{x}_i(q)\boldsymbol{\beta} = 0$ then by (3) $H\boldsymbol{\beta} = \boldsymbol{0}$, which, together with the invertibility of H, implies that $\boldsymbol{\beta} = \boldsymbol{0}$. But $\boldsymbol{\beta} \neq \boldsymbol{0}$ by assumption. Therefore if $F_l A_l$ is a singular matrix, we have

$$1 + \boldsymbol{x}_i(q)\boldsymbol{\gamma}_l = 0.$$

The preceding argument implies that $F_l A_l$ is nonsingular if

$$1 + \boldsymbol{x}_i(q)\boldsymbol{\gamma}_l \neq 0. \tag{4}$$

A randomized algorithm and the existence of a solution

Let us summarize the analysis up to this point. The coding vectors $\boldsymbol{x}_i(q), q \in Q_i$, can be assigned in a way that meet our requirements if

$$\tau \triangleq \prod_{t_l : q \in Q_i^l[j], j \in \{1,\cdots,m_i\}} (1 + (\boldsymbol{x}_i(q) - \boldsymbol{y}_i(p_l))\boldsymbol{\gamma}_l) \prod_{t_l : q \notin Q_i^l[j], j \in \{1,\cdots,m_i\}} (1 + \boldsymbol{x}_i(q)\boldsymbol{\gamma}_l) \neq 0.$$

In the preceding equation t_l is restricted to the destinations for which $K_l \geq i+1$, and the vectors $\boldsymbol{\gamma}_l$ and $\boldsymbol{y}_i(p_l)$ are specified in the analyses of Cases 1 and 2.

One other constraint is that $\boldsymbol{x}_i(q)$ for $q \in Q_i[j]$ can only be a linear combination of the vectors $\{\boldsymbol{y}_i(p) : p \in P_i[j]\}$. Let us choose each $\boldsymbol{x}_i(q)$ to be a random linear combination of the elements of $\{\boldsymbol{y}_i(p) : p \in P_i[j]\}$ where the coefficient of each $\boldsymbol{y}_i(p)$ is randomly and independently chosen from the uniform distribution over the field $\mathbb{F}$. For each destination t_l with $K_l \geq i+1$ define ϕ_l as the event that the corresponding term in the product above is zero. Then we have

$$\Pr(\tau = 0) = \Pr(\bigvee_{t_l : K_l \geq i+1} \phi_l) \leq \sum_{t_l : K_l \geq i+1} \Pr(\phi_l).$$

Suppose that $q \in Q_i[j]$ and $\boldsymbol{x}_i(q) = \sum_{p \in P_i[j]} \theta_p \boldsymbol{y}_i(p)$. Now consider a destination t_l with $K_l \geq i+1$. If $q \in Q_i^l[j]$ and $p_l \in P_i^l[j]$ is matched with q, we need to have $1 + (\boldsymbol{x}_i(q) - \boldsymbol{y}_i(p_l))\boldsymbol{\gamma}_l \neq 0$. There exist $\omega_0 \in \mathbb{F}, \omega_p \in \mathbb{F}, p \in P_i[j]$, which are determined by $\boldsymbol{y}_i(p)$ and $\boldsymbol{\gamma}_l$ and satisfy

$$1 + (\boldsymbol{x}_i(q) - \boldsymbol{y}_i(p_l))\boldsymbol{\gamma}_l = \omega_0 + \sum_{p \in P_i[j]} \omega_p \theta_p$$

There are two cases to consider. First, if $\omega_p = 0$ for all $p \in P_i[j]$, then $\omega_0 + \sum_{p \in P_i[j]} \omega_p \theta_p = \omega_0$ is a constant independent of $\theta_p, p \in P_i[j]$. Furthermore by setting $\theta_{p_l} = 1$ and $\theta_p = 0$ for $p \in P_i[j]$ and $p \neq p_l$ so that $\boldsymbol{x}_i(q) = \boldsymbol{y}_i(p_l)$, we find that

$$\omega_0 = 1 + (\boldsymbol{x}_i(q) - \boldsymbol{y}_i(p_l))\boldsymbol{\gamma}_l = 1.$$

Therefore in this case $\Pr(\phi_l) = 0$. Next if there exists some $p \in P_i[j]$ for which $\omega_p \neq 0$ then $\omega_0 + \sum_{p \in P_i[j]} \omega_p \theta_p$ depends on $\theta_p, p \in P_i[j]$. Since $\theta_p, p \in P_i[j]$, are uniformly distributed random variables over $\mathbb{F}$, $\omega_0 + \sum_{p \in P_i[j]} \omega_p \theta_p$ is likewise uniformly distributed over $\mathbb{F}$. In this case $\Pr(\phi_l) = \frac{1}{|\mathbb{F}|}$.

Next suppose that $q \notin Q_i^l[j]$. Here we need to have $1 + \boldsymbol{x}_i(q)\boldsymbol{\gamma}_l \neq 0$. Following the preceding argument, there exist $\omega_0 \in \mathbb{F}, \omega_p \in \mathbb{F}, p \in P_i[j]$, which are determined by $\boldsymbol{y}_i(p)$ and $\boldsymbol{\gamma}_l$ and satisfy

$$1 + \boldsymbol{x}_i(q)\boldsymbol{\gamma}_l = \omega_0 + \sum_{p \in P_i[j]} \omega_p \theta_p.$$

If $\omega_p = 0$ for all $p \in P_i[j]$, then $\omega_0 + \sum_{p \in P_i[j]} \omega_p \theta_p = \omega_0$ is a constant independent of $\theta_p, p \in P_i[j]$. By setting $\theta_p = 0$ for all $p \in P_i[j]$ so that $\boldsymbol{x}_i(q) = \mathbf{0}$, we obtain $\omega_0 = 1 + \boldsymbol{x}_i(q)\boldsymbol{\gamma}_l = 1$, and $\Pr(\phi_l) = 0$. If there is some $p \in P_i[j]$ for which $\omega_p \neq 0$, then an analogous argument to our earlier one implies that $\Pr(\phi_l) = \frac{1}{|\mathbb{F}|}$.

As a result, for any destination t_l with $K_l \geq i+1$, we have $\Pr(\phi_l) \leq \frac{1}{|\mathbb{F}|}$. Therefore

$$\Pr(\tau = 0) \leq \sum_{t_l : K_l \geq i+1} \Pr(\phi_l) \leq \frac{g}{|\mathbb{F}|}.$$

Since we are interested in the event that $\tau \neq 0$, we have

$$\Pr(\tau \neq 0) \geq 1 - \frac{g}{|\mathbb{F}|}.$$

Therefore if $|\mathbb{F}| > g$, then $\Pr(\tau \neq 0) > 0$ and there is at least one valid solution for $\boldsymbol{x}_i(q)$. This also yields a randomized algorithm with probability of success of at least $1 - \frac{g}{|\mathbb{F}|}$. If we take the size of the field to be $|\mathbb{F}| \geq 2g$ then the probability of success will be at least $1 - \frac{g}{2g} = \frac{1}{2}$ for each $q \in Q_i$.

A deterministic polynomial time algorithm

We next explain a deterministic algorithm with polynomial running time for the finding vectors $\boldsymbol{x}_i(q), q \in Q_i[j]$. For each $q \in Q_i$ we seek a vector $\boldsymbol{u} = \boldsymbol{x}_i(q)$ which is a linear combination of the vectors in $\{\boldsymbol{y}_i(p) : p \in P_i[j]\}$ such that for any destination t_l with $K_l \geq i+1$, if $q \in Q_i^l[j]$ and $p_l \in P_i^l[j]$ is matched with q, then $1 + (\boldsymbol{u} - \boldsymbol{y}_i(p_l))\boldsymbol{\gamma}_l \neq 0$, and if $q \notin Q_i^l[j]$ then $1 + \boldsymbol{u}\boldsymbol{\gamma}_l \neq 0$.

Define the subset of indices of destinations W as

$$W = \Big\{ l \in \{1, \cdots, g\} : K_l \geq i+1, q \in Q_i^l[j] \text{ for some } j \in \{1, \cdots, m_i\}, \boldsymbol{y}_i(p_l)\boldsymbol{\gamma}_l \neq 0 \Big\}.$$

We can write the conditions that $\boldsymbol{u}$ needs to satisfy as $1 + (\boldsymbol{u} - \boldsymbol{y}_i(p_l))\boldsymbol{\gamma}_l \neq 0$ for $l \in W$ and $1 + \boldsymbol{u}\boldsymbol{\gamma}_l \neq 0$ for $l \notin W$ and $K_l \geq i+1$. Next we use [7, Lemma 8]:

Lemma 2. *Let $n \leq |\mathbb{F}|$. Let $\boldsymbol{a}_1, \cdots, \boldsymbol{a}_n \in \mathbb{F}^{1\times R}$ and $\boldsymbol{b}_1, \cdots, \boldsymbol{b}_n \in \mathbb{F}^{R\times 1}$ with $\boldsymbol{a}_i\boldsymbol{b}_i \neq 0, i \in \{1, \cdots, n\}$. There exists a linear combination $\boldsymbol{c}$ of $\boldsymbol{a}_1, \cdots, \boldsymbol{a}_n$ such that $\boldsymbol{c}\boldsymbol{b}_i \neq 0, i \in \{1, \cdots, n\}$. Such a vector $\boldsymbol{c}$ can be found in time $O(n^2R)$.*

If we are given the set of vectors $\boldsymbol{\gamma}_l$, then it takes $O(gR)$ steps to form the set W. Then by applying the preceding lemma, if $g \leq |\mathbb{F}|$, we can find a vector $\boldsymbol{w} \in \mathbb{F}^{1\times R}$ such that $\boldsymbol{w}$ is a linear combination of the vectors in $\{\boldsymbol{y}_i(p_l) : l \in W\}$ and for every $l \in W$, we have that $\boldsymbol{w}\boldsymbol{\gamma}_l \neq 0$. Furthermore vector $\boldsymbol{w}$ can be found in time $O(g^2R)$. By adding the time $O(gR)$ needed to produce set W, we need a total time of $O(g^2R + gR) = O(g^2R)$ to find vector $\boldsymbol{w}$. Next we let $\boldsymbol{u} = \sigma\boldsymbol{w}$ for some $\sigma \in \mathbb{F}$. We show that an appropriate value of σ exists such that $\boldsymbol{u}$ satisfies all of the constraints.

For $l \in W$, we need to have $1 + (\sigma\boldsymbol{w} - \boldsymbol{y}_i(p_l))\boldsymbol{\gamma}_l \neq 0$. Therefore

$$\sigma \neq \frac{\boldsymbol{y}_i(p_l)\boldsymbol{\gamma}_l - 1}{\boldsymbol{w}\boldsymbol{\gamma}_l}. \tag{5}$$

For $l \notin W$ and $K_l \geq i+1$ we need to have $1 + \sigma\boldsymbol{w}\boldsymbol{\gamma}_l \neq 0$. If $\boldsymbol{w}\boldsymbol{\gamma}_l = 0$ then this condition is fulfiled for all values of σ. Otherwise we need to have

$$\sigma \neq \frac{-1}{\boldsymbol{w}\boldsymbol{\gamma}_l}. \tag{6}$$

There are at most g constraints of the form (5) and (6) on σ. Therefore if the size of field $\mathbb{F}$ is greater than the number of destinations g, this deterministic approach will find at least one σ that is not in the discriminating set by at most considering g elements of $\mathbb{F}$. Therefore the total complexity of finding an appropriate value of σ is $O(g)$ and the total complexity of finding vector $\boldsymbol{u}$ is $O(g + g^2R) = O(g^2R)$.

To find the overall complexity of finding the vector $\boldsymbol{x}_i(q)$, we need to evaluate the complexity of finding vector $\boldsymbol{\gamma}_l$ for every $l \in \{1, \cdots, g\}$ with $K_l \geq i+1$. From the analysis of Cases 1 and 2, $\boldsymbol{\gamma}_l = H^{-1}\boldsymbol{\alpha}$, where matrix H is F_lA_l from the previous step of the algorithm. Since matrix F_l has size $R \times L$ and matrix A_l has size $L \times R$ for some $R \leq L \leq |Q_i|$, computing H needs $O(R|Q_i|)$ operations. Evaluating H^{-1} also needs $O(R^3)$ steps and so there are a total of $O(R|Q_i| + R^3)$ operations for evaluating $\boldsymbol{\gamma}_l$. Since there are at most g different $l \in \{1, \cdots, g\}$ with $K_l \geq i+1$, we will have $O(gR|Q_i| + gR^3)$ as the total complexity of evaluating different values of $\boldsymbol{\gamma}_l$ for any specific $q \in Q_i$. Therefore the total complexity of evaluating $\boldsymbol{x}_i(q)$ will be $O(gR|Q_i| + gR^3 + g^2R)$.

Let us assume that the number of nodes m_i at each layer $i \in \{1, \cdots, M\}$ is at most m. Furthermore assume that the size of transmitted and received signals at each node is at most r. Therefore the total complexity of evaluating each $\boldsymbol{x}_i(q)$ will be $O(gRmr + gR^3 + g^2R)$. Since there are at most

mMr different $\boldsymbol{x}_i(q)$ to be evaluated, if we assume that the unicast flows from source to each destination is provided, the total complexity of our algorithm is $O(gRm^2Mr^2+gR^3mMr+g^2RmMr) = O(gRmMr(mr+R^2+g))$.

The complexity of computing a unicast flow to a destination by the algorithm given in [6] is $O(M(mr)^3 \log mr)$. Since we have g destinations, the total complexity of computing the unicast flows will be $O(gM(mr)^3 \log mr)$. If we add this running time to the running time of our algorithm, the total running time will be $O(gmMr(mrR+R^3+gR+(mr)^2 \log mr))$. We can compare it to the running time of the algoithm given in [4] which is $O(g(r^2mM+R)^3 \log(r^2mM+R)+r^2mM(r^2mM+R)^2+(g \log gRM)^3)$ and see that our algorithm is considerably faster.

Number of network uses to achieve capacity

We have shown that it is sufficient for the size of the field of operation $\mathbb{F}$ of the LDRN to be greater than g to guarantee the existence of a multicast coding solution. In general however, the network operates over some fixed field which is usually $\mathbb{F}_p$ for some prime number p. In order to achieve a greater field size, we will use multiple rounds of the network. Here we will argue that if we use the network for k rounds, it is equivalent to an LDRN with field of operation $\mathbb{F} = \mathbb{F}_p^k$. This implies that in order to have a field size at least $g+1$, it is sufficient to use the network for $k = \lceil \log_p(g+1) \rceil$ rounds. This is an improvement over the number of rounds that is needed for the algorithm of [4] which is $k \approxeq \log_p(g(\log_p g - 1)RM)$.

Suppose that the network is used for k rounds and we use the superscript $t \in \{0, \cdots, k-1\}$ to denote the time index that a vector is received or sent. For each $i \in \{1, \cdots, M-1\}$ we have

$$\mathbf{y}_{i+1}^t = G_i \mathbf{x}_i^t, \qquad t \in \{0, \cdots, k-1\}$$

Observe we can use a dummy variable D as the unit delay operator and represent the preceding k equations as a single equation

$$\sum_{t=0}^{k-1} \mathbf{y}_{i+1}^t D^t = G_i \sum_{t=0}^{k-1} \mathbf{x}_i^t D^t.$$

Next, notice that $\sum_{t=0}^{k-1} \mathbf{y}_{i+1}^t D^t$ and $\sum_{t=0}^{k-1} \mathbf{x}_i^t D^t$ can be regarded as new vectors in the extension field $\mathbb{F}_p^k$ and we can assume that the network is operating in the extension field $\mathbb{F}_p^k$. Since the transfer matrix between the layers i and $i+1$ is still G_i and has not changed in the new field, the existence of the unicast flow over the original field implies the existence of flow over the extended field. Therefore our analysis is valid over any field $\mathbb{F}_p^k$.

References

[1] A. Amaudruz and C. Fragouli, “Combinatorial algorithms for wireless information flow,” in *Proceedings 2009 ACM-SIAM Symposium on Discrete Algorithms (SODA 2009)*, New York, January 2009.

[2] A. S. Avestimehr, S. N. Diggavi, and D. N. C. Tse, “Wireless network information flow: A deterministic approach,” *IEEE Transactions on Information Theory*, to appear.

[3] A. S. Avestimehr, S. N. Diggavi, and D. N. C. Tse, “Approximate capacity of Gaussian relay networks,” in *Proceedings of 2008 IEEE International Symposium on Information Theory (ISIT)*, 2008, pp. 474 –478.

[4] J. Ebrahimi, C. Fragouli, "Vector network coding algorithms", *Proceedings of IEEE Symposium on Information Theory (ISIT2010),* Austin, June 2010.

[5] E. Erez, Y. Xu and E. M. Yeh, "Coding for the deterministic network model," *Information Theory and Applications Workshop (ITA 2010),* San Diego, Feb. 2010.

[6] M. Goemans, S. Iwata, and R. Zenklusen, "An algorithmic framework for wireless information flow," in *47th Annual Allerton Conference on Communication, Control, and Computing ,* 2009, pp. 294–300.

[7] S. Jaggi, P. Sanders, P. A. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen, "Polynomial Time Algorithms for Multicast Network Code Construction," *IEEE Transactions on Information Theory*, vol. 51, no. 6, pp. 1973–1982, 2005.

[8] M. Khojastepour and A. Keshavarz-Haddad, "Rotational coding achieves multicast capacity of deterministic wireless networks," in *47th Annual Allerton Conference on Communication, Control, and Computing.* 2009, pp. 1330–1337.

[9] M. Kim and M. Médard, "Algebraic network coding approach to deterministic wireless relay networks," in *Proceedings Allerton Conference on Communications, Control, and Computing,* Monticello, IL, USA, Sept. 2010.

[10] R. Koetter, M. Médard, "An algebraic approach to network coding," *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, pp. 782–795, Oct. 2003.

[11] S. M. S. Tabatabaei Yazdi and S. A. Savari, "A max-flow/min-cut algorithm for a class of wireless networks," in *ACM-SIAM Symposium on Discrete Algorithms (SODA 2010),* Austin, Texas, Jan 2010, arxiv.org e-print 0904.2401, April 2009.

[12] S. M. S. Tabatabaei Yazdi and S. A. Savari, "A combinatorial study of linear deterministic relay networks," in *Proceedings 2010 Information Theory Workshop (ITW)*, Cairo, Egypt, Jan 2010, pp. 1–5.